

# Primera actividad de laboratorio

Redes de Telecomunicaciones/Computadoras

10/sep/2019

## 1. Trazo de gráficas en 2D

### 1.1. Descripción

En este curso, utilizaremos trazas de eventos para estudiar diferentes procesos que ocurren en una red. Una *traza* es un archivo (de texto normalmente) que registra la dinámica de algún proceso, e.g., el retardo de extremo a extremo, el *throughput*, las pérdidas, la variabilidad del retardo (*jitter*), por mencionar algunos. Evitaremos usar hojas de cálculo, y utilizaremos en cambio herramientas de cálculo numérico (e.g., Matlab, Octave, Scilab) o simplemente de trazo de gráficas (e.g., GNUPlot, Grace). La intención de esta actividad es entonces, realizar ejercicios básicos de trazo de gráficas a partir de trazas. Las trazas que usaremos describen diferentes procesos que ocurren en Internet. Para proceder con la realización de esta actividad, deberá entonces contar con algunas de las herramientas que se acaban de mencionar instaladas en su sistema.

La traza 1, corresponde a la dinámica de la ventana de congestión del protocolo TCP (*Transmission Control Protocol*) durante un intervalo de tiempo. La ventana de congestión es el principal parámetro del protocolo TCP para determinar la cantidad de segmentos de datos que el protocolo inyecta en la red a cada instante de tiempo. Esto lo estudiaremos más adelante en el curso.

La traza 2, corresponde al retardo de extremo a extremo de paquetes enviados durante una conversación de telefonía en Internet (VoIP, *Voice over IP*). El retardo de extremo a extremo es un parámetro determinante en el desempeño de cualquier aplicación interactiva en Internet. Este fenómeno lo estudiaremos ampliamente durante el curso. Esta traza muestra algunos paquetes que forman parte de un par de frases de la conversación registrada, no a la sesión completa.

Finalmente, la traza 3 corresponde al retardo de ida y vuelta (RTT, *Round Trip Time*) de una sesión en Internet. El retardo de ida y vuelta es el tiempo transcurrido desde que un paquete se envía en el transmisor hasta que éste recibe un acuse de recibo de la entidad receptora. El RTT es un factor fundamental que determina el tiempo que un emisor TCP espera por un acuse de recibo (ACK, *acknowledgment*) en el protocolo TCP.

### 1.2. Experimentación

#### 1.2.1. Trazo 1: Ventana de congestión de TCP

Descargue la traza 1 en su sistema local de archivos y trace las gráficas de la Figura 1. La gráfica de la izquierda puede trazarse en Octave con los comandos de abajo. Para configurar el aspecto de la línea y del marcador consulte esta parte de la documentación de Matlab (con el cual, es compatible Octave). En el ejemplo de abajo, '- -r' indica que se desea una gráfica con líneas semiocultas (- -), sin marcador, y que el trazo sea de color rojo (r).

```
load tcpplot2.dat
plot(tcpplot2, '--r', 'LineWidth', 3)
grid on
xlabel('Vuelta de transmisión')
ylabel('cwnd [segmentos]')
```

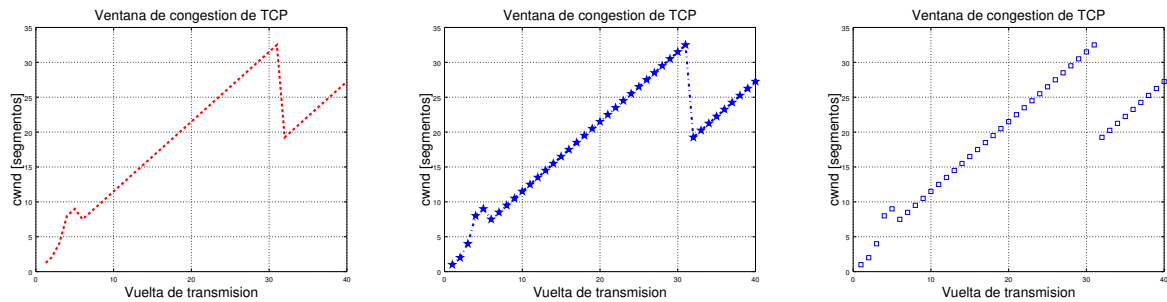


Figura 1: Dinámica de la ventana de congestión de TCP Reno.

```
title('Ventana de congestión de TCP')
print -dpng "traza1a.png"
```

Por otro lado, usando GNUPlot se podrían emplear las siguientes líneas de código:

```
set term png
set output "traza1a.png"
set xlabel 'Vuelta de transmision'
set ylabel 'cwnd [segmentos]'
set title 'Ventana de congestion de TCP'
plot "tcpplot2.dat" with lp linetype 3
```

### 1.2.2. Traza 2: retardo de extremo a extremo

Descargue la traza 2 para realizar las gráficas de la Figura 2. Octave llama *vástagos* (*stem*) al tipo de la figura izquierda, mientras que GNUPlot le llama *impulsos* (*impulses*). Discuta sobre la utilidad de cada tipo de gráfica para observar el mismo proceso.

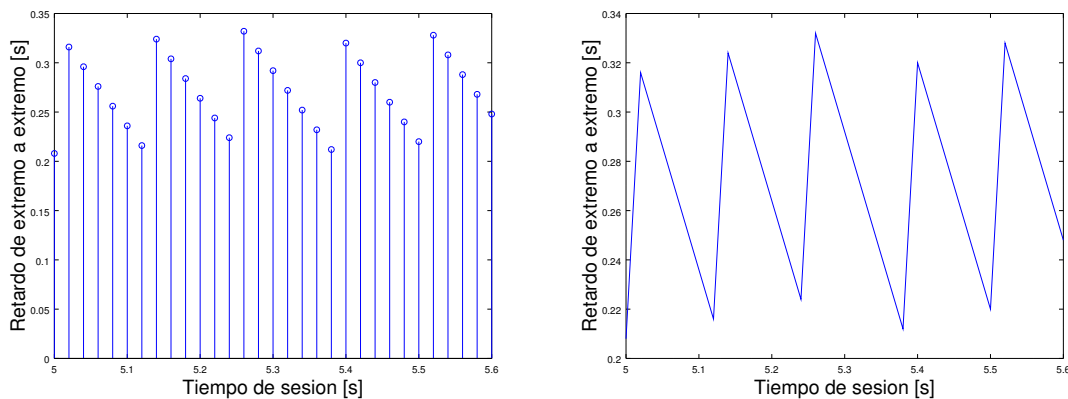


Figura 2: Retardo de extremo a extremo de paquetes de voz sobre Internet (VoIP).

### 1.2.3. Traza 3: RTT

Finalmente, descargue la traza 3 y realice la gráfica de la Figura 3. Observe los límites inferior y superior en el eje de las  $x$ . Para obtener esta gráfica, deberá entonces buscar la función para ajustar los límites de los ejes.

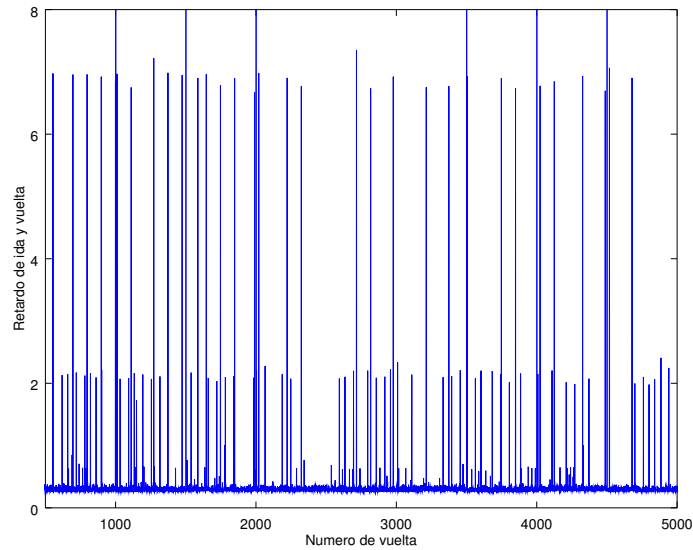


Figura 3: Tiempo de ida y vuelta [s].

### 1.3. Documentación

Octave es ampliamente compatible con Matlab, así que el ejemplo para Octave mostrado arriba funciona igualmente en Matlab. Por otro lado, tanto Octave como GNUPlot pueden instalarse en múltiples sistemas operativos como Windows, MacOS y Linux. En Matlab Primer se describen los fundamentos del uso de Matlab. En particular, los ejemplos de trazo de gráficas que realizamos en el laboratorio están bien documentados. Estos mismos ejemplos funcionan también con Octave. Por otro lado, podrá encontrar la documentación de GNUPlot aquí.